Optimizing Winning Strategies in Regular Infinite Games

Wolfgang Thomas

RWTH Aachen, Lehrstuhl Informatik 7, 52056 Aachen, Germany
thomas@informatik.rwth-aachen.de

Abstract. We consider infinite two-player games played on finite graphs where the winning condition (say for the first player) is given by a regular omega-language. We address issues of optimization in the construction of winning strategies in such games. Two criteria for optimization are discussed: memory size of finite automata that execute winning strategies, and – for games with liveness requests as winning conditions – "waiting times" for the satisfaction of requests. (For the first aspect we report on work of Holtmann and Löding, for the second on work of Horn, Wallmeier, and the author.)

1 Introduction

A central result in the algorithmic theory of infinite two-player games is the Büchi-Landweber Theorem [2]. It shows how to "solve" finite-state games where the winning condition is given by a regular ω-language (over the state space of the underlying finite game arena). The solution of a game involves two algorithms, the first one to decide for each state who of the two players wins the plays starting from this state, and the second one to synthesize a winning strategy for the respective winner. As it turns out, finite-state winning strategies suffice, i.e. strategies that are realized by a finite-state machine with output (for example in the format of a Mealy automaton). For some further background see [10, 13].

The Büchi-Landweber Theorem provides an approach for the automatic synthesis of finite-state controllers, assuming that the synthesis problem can be modeled in the framework of finite-state games (with a regular winning condition). Two directions of research have been opened by this fundamental result. First, various options of extending the framework have been investigated, such as infinite-state games, concurrent games, distributed (or multiplayer) games, and stochastic games, and in each case the scope of models that still allow an algorithmic solution has been explored. The other direction aims at a refined analysis of finite-state games, with the objective to single out cases that are interesting in applications and allow an efficient treatment. We discuss the second problem in this paper, starting with the observation that from a practical viewpoint the Büchi-Landweber Theorem is insufficient in two respects: It involves procedures of high computational complexity, and – if a controller exists – the controller's behavior is not optimized in any way. Progress on both questions seems possible

since the full power of regular winning conditions is rarely needed in applications; many interesting cases can be studied in a more restricted framework. As important conditions of this kind we mention (and pursue below) the "weak" winning conditions and the "request-response conditions". In this context we address the aspect of optimization for two criteria:

1. memory size of the controller
2. waiting times when liveness conditions are to be satisfied.

Regarding these criteria we report (in Sections 2 and 3, respectively) on recent and initial results obtained in the Aachen research group. The results on memory reduction are due to Michael Holtmann and Christof Löding [7, 6], the results on minimizing waiting times in request-response games to Florian Horn, Nico Wallmeier, and the author [11, 8].

2 Reducing Memory in Winning Strategies

It is well-known that regular winning conditions are reducible to a format that is called "Muller condition". A game with a Muller winning condition is given by a pair $(G, \mathcal{F})$ where G is a finite game arena (or: game graph), say, over the set Q, and $\mathcal{F}$ is a family of subsets of Q. A play $\varrho \in Q^\omega$ is declared as won by player 0 (playing against player 1) according to the Muller condition if the set $\mathrm{Inf}(\varrho)$ of states visited infinitely often in ϱ belongs to $\mathcal{F}$. As shown in [3], a lower bound for the memory size of finite-state winning strategies in Muller games is given by the factorial function (in the number of vertices of the game graph). An exponential lower bound is known for the so-called weak Muller games, which are also presented as pairs $(G, \mathcal{F})$ and in which a play is won by player 0 if the "occurrence set" $\mathrm{Occ}(\varrho)$ of states visited in ϱ belongs to $\mathcal{F}$.

The known solutions of (Muller and weak Muller) games yield finite-state strategies as Mealy automata that can then be minimized by classical techniques. A serious disadvantage of this method is the fact that the initial Mealy automaton is not constructed in a way that supports memory reduction. There are examples where a given finite-state winning strategy is minimal (i.e., cannot be reduced w.r.t. number of states), but where another winning strategy exists with exponentially less states. The main (and still open) problem is to have an overview of all possible winning strategies and a method to single out the "small" ones.

In their recent work [7, 6] Holtmann and Löding developed a technique which achieves partial progress in the search for memory-optimal strategies. The main idea is to do some preprocessing of the game arena – reducing it in size – so that the subsequent application of the standard algorithms leads to winning strategies with smaller memory.

An efficient preprocessing turns out to be possible for the case of weak Muller games. We describe the technique for this example; other cases such as Muller games and Streett games are treated in ongoing work (including experimental studies).

The first step in the algorithm is to reduce a weak Muller game to a weak parity game. (The weak parity condition refers to a coloring of states by natural numbers, and it requires that the highest used color in a play is even.) This reduction involves an expansion of the state space of the given arena G to a new arena G': From the state set Q of G we proceed to $2^Q \times Q$. Intuitively, each play ϱ over Q is mapped to a new play ϱ' over $2^Q \times Q$ where, at each moment t, $\varrho'(t)$ is the pair consisting of (1) the set of previously visited states in ϱ and (2) the current state $\varrho(t)$. The first component is thus also called the "appearance record" (of visited states). An important property of the appearance record over Q is the fact that for each weak Muller condition over Q one can build a finite-state winning strategy with appearence records as memory states (if a winning strategy exists at all).

The second (and essential) step is a minimization of the resulting game graph G' over $2^Q \times Q$ with the weak parity winning condition. The game graph G' is converted into a weak parity automaton $\mathcal{A}$ (with state space $2^Q \times Q$ and input alphabet Q) accepting precisely those plays ϱ over G that satisfy the given weak Muller condition. A variant of the Löding's efficient minimization of weak automata [9] yields a reduced automaton $\mathcal{A}'$ which accepts the same plays as $\mathcal{A}$; this automaton is then converted back into a weak parity game (over a quotient graph G'' of G'). The specific structure of game graphs (with vertices attributed to the two players, which deviates from the format of acceptors) requires some technical work in order to be able to switch from games to automata and back. In particular, the form of the state equivalence that leads to a state space reduction has to be designed appropriately.

The approach of optimizing winning strategies via a reduction of game graphs can result in an exponential improvement of the size of finite-state strategies; a family of examples is given in [6].

The main point for the practical applicability of this approach is the availability of efficient minimization (or reduction) procedures for ω-automata. In the case of deterministic weak parity automata, as needed above, the minimization algorithm of [9] can be used. For more general winning conditions, such as the Büchi condition and the (strong) parity condition, other algorithms have to be used ([4],[5]; see [6]).

Many questions remain open in this field. Already in [2], Büchi and Landweber raise the question whether the space of all (winning) strategies for a given Muller game can be parametrized by data that are derived from the loop structure of a Muller game graph. We do not have as yet any transparent scheme that captures all winning strategies of an infinite game. Another problem is motivated by the fact that many game specifications are given as logical formulas (e.g., formulas of linear-time temporal logic LTL). In this case, the first step is to construct a Muller game which is then treated as discussed above. The interplay between these two steps (introducing states for the Muller game, and introducing further states for its solution in terms of finite-state strategies) is not well understood; it seems reasonable to expect gains in efficiency by integrating these two levels of state space reduction.

3 Minimizing Waiting Times in Liveness Conditions

Many specifications (winning conditions) for infinite games consist of a combination of a safety condition ("all states visited in play should share a certain "good" state property") with liveness conditions of the following form:

($*$) Whenever a P-state is visited, later also an R-state is visited

where P and R are subsets of the vertex set of the game graph (state properties). This corresponds to a situation where a visit to P signals a certain "request" and a visit to R the corresponding "response" (usually meaning that satisfaction of the request is granted). The safety conditions can be captured by a restriction of the game graph. We thus concentrate on the second type of winning condition ($*$). A *request-response condition* is a finite conjunction of conditions of the form ($*$). In linear-time temporal logic this amounts to a conjunction $\bigwedge_{i=0}^{k} \mathrm{G}(p_i \rightarrow \mathrm{XF}\ r_i)$ with formulas p_i, r_i expressing state properties. A request-response game (in short, RR-game) is a finite-state game with a request-response winning condition.

It is not difficult to show that RR-games can be reduced to Büchi games, in which the winning condition just requires to visit states in a certain designated set F infinitely often [12]. Via this reduction (and the solution of Büchi games), the RR-games can be solved, and a finite-state winning strategy can be constructed for the respective winning player. In most practical scenarios, however, the construction of a finite-state strategy which just ensures winning an RR-game is unsatisfactory. As in scheduling problems, one would like to have a solution that minimizes the time lags between visits to a set P and the succeeding visits to R afterwards (referring to a condition ($*$)). An approach that is familiar from "parametrized model-checking" [1] is to bound the time lags by a constant and to check whether this bound is respected for all RR-conditions involved. A corresponding optimization problem asks to compute the minimal possible bound. We pursue here a more ambitious goal, namely to realize an optimization of the waiting times for the individual RR-conditions. Furthermore, we use real numbers as values for optimization; they reflect average waiting times in infinite plays. We obtain associated valuations for winning strategies, and declare a winning strategy as optimal if it realizes the optimal value (if it exists) among all winning strategies.

We discuss here two natural valuations (a more general framework is treated in [8]). Let us first consider just a single RR-condition with sets P, R as above, and assume that a play ϱ satisfies the condition. We count the time lags between "first visits" to P, which we call "activations of waiting" and the successive R-visits. (A visit is "first" if all previous visits to P are already matched by an R-visit.) For each finite play prefix $\varrho[0 \ldots n]$, the average waiting time $w_{\varrho,0}(n)$ is the sum of the waiting times in the prefix $\varrho[0 \ldots n]$ divided by the number of activations.

A weakness of this approach of "linear penalty for waiting" is that it does not distinguish between the following two kinds of plays, with two RR-conditions to

be observed: In the first play, the conditions are met with waiting times 9 and 1, respectively, in the second both conditions are satisfied with waiting time 5 each (and the repetition is assumed to be the same in both cases). The intuitive preference for the second solution (where the maximum waiting time is smaller) is met by introducing the penalty m (rather than 1) for the m-th successive time instance of waiting. This results in a quadratic increase of the penalty during a waiting phase; and we define $w_{\varrho,1}(n)$ as the sum of these penalties for the play prefix $\varrho[0 \dots n]$, again divided by the number of activations.

Let us write $w_\varrho(n)$ for any of these two values $w_{\varrho,i}(n)$. (We suppress the index $i \in \{0,1\}$ for better readability.) The value of a play ϱ is then $v(\varrho) := \limsup_{n\to\infty} w_\varrho(n)$ (which is set to be ∞ if this limit does not exist). If there are k RR-conditions with respective values $w^j_\varrho(n)$ $(j = 1, \dots, k)$, then the value of ϱ is defined as

$$v(\varrho) := \limsup_{n\to\infty} \frac{1}{k} \sum_{j=1}^{k} w^j_\varrho(n)$$

We write $v(\sigma, \tau)$ for the value of the play that is determined by the strategies σ, τ of players 0 and 1, respectively. The value of strategy σ is $v(\sigma) := \sup_\tau v(\sigma, \tau)$. A winning strategy for player 0 is called optimal if there is no other winning strategy for player 0 with smaller value.

A simple example shows that the value derived from linear penalty for waiting does not allow, in general, to construct optimal winning strategies that are finite-state. However, in the case of quadratic penalty for waiting, an optimal finite-state winning strategy exists, which moreover can be effectively computed ([11, 8]). The key fact to establish this result is a lemma that ensures a uniform bound on the waiting times for an optimal winning strategy. An optimal finite-state strategy can then be computed via a reduction to the solution of mean-payoff games [14].

This study addresses games where the payoff of a play is different from 0 or 1. This view is standard, for example, in mean-payoff games and stochastic games. Here we apply it to discrete infinite games, extract continuous parameters, and referring to them compute again discrete optimal objects (finite-state strategies).

4 Conclusion

We discussed two problems of optimization in solving infinite games with regular winning conditions. While the two problems are quite different in nature, they point to a shift in the analysis of games. The origins of the theory of infinite games are descriptive set theory and questions of mathematical logic, in which the existence of winning strategies is the main issue. The refined studies in theoretical computer science (and their applications in system design) naturally lead to modifications of the objectives. The present paper illustrates this by studies on quantitative classifications of winning strategies.

References

1. R. Alur, K. Etessami, S. La Torre, D. Peled. Parametric temporal logic for "model measuring", *ACM Trans. Comput. Logic* 2 (2001), 388-407.
2. J.R. Büchi, L.H. Landweber. Solving sequential conditions by finite-state strategies, *Trans. Amer. Math. Soc.* 138 (1969), 367-378.
3. S. Dziembowski, M. Jurdziński, I. Walukiewicz. How much memory is needed to win infinite games?, *Proc. 12th IEEE Symp. on Logic in Computer Science*, IEEE Computer Society Press 1997, pp. 99-110.
4. K. Etessami, Th. Wilke, R.A. Schuller. Fair bismulation relations, parity games, and state space reduction for Büchi automata, *SIAM J. Comput.* 34 (2005), 1159-1175.
5. C. Fritz, Th. Wilke. Simulation relations for alternating parity automata and parity games, *Proc. 10th DLT*, Spinger LNCS 4036 (2006), 59-70.
6. M. Holtmann, C. Löding. Memory reduction for strategies in infinite games, Proc. CIAA 2007, Springer LNCS 2007.
7. M. Holtmann, *Memory Reduction for Strategies in Infinite Games*, Diploma thesis, RWTH Aachen 2007. www.automata.rwth-aachen.de/∼holtmann/
8. F. Horn, W. Thomas, N. Wallmeier, Optimal strategy synthesis for request-response games, submitted.
9. C. Löding. Efficient minimization of deterministic weak ω-automata, *Inf. Proc. Lett.* 79 (2001), 105-109.
10. W. Thomas. On the synthesis of strategies in infinite games, *Proc. STACS 1995*, Springer LNCS 900 (1995), 1-13.
11. N. Wallmeier. *Strategien in unendlichen Spielen mit Liveness-Gewinnbedingungen: Syntheseverfahren, Optimierung und Implementierung*, Dissertation, RWTH Aachen 2007.
12. N. Wallmeier, P. Hütten, W. Thomas. Symbolic synthesis of finite-state controllers for request-response specifications, Proc. 8th CIAA, Springer LNCS 2759 (2003), 11-22.
13. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees, *Theor. Comput. Sci.* 200 (1998), 135-183.
14. U. Zwick, M. Paterson. The complexity of mean payoff games on graphs, *Theor. Comput. Sci.* 158 (1996), 343-259.